Journal of Physics: Conference Series

PAPER • **OPEN ACCESS**

Human machine interface (HMI) based on a multi-agent system in a water purification plant

To cite this article: E. Mendoza *et al* 2021 *J. Phys.: Conf. Ser.* **2090** 012122

Human machine interface (HMI) based on a multi-agent system in a water purification plant

E. Mendoza[1*], J. Andramuño[1], J. Núñez [2], L. Córdova [1]

[1] Automation Department. Universidad Católica de Santiago de Guayaquil, Ecuador
[2] Department of Energy. Universidad de la Costa, Colombia

[*]Corresponding author: eduardo.mendoza01@cu.ucsg.edu.ec

Abstract. The applications of multi-agent systems (MAS) are growing increasingly in the industrial field due to the advantages inherent to their characteristics and properties, the use of distributed automation architectures, which have satisfactorily solved control problems that its complexity and dynamic behavior have not been properly resolved with other approaches under these conditions, intelligent agents must meet the requirements of current automation systems, such as autonomy, flexibility, reconfiguration, in concurrent and collaborative systems, which traditionally do not have been designed to satisfy these characteristics. In the present work, a distributed architecture is proposed for the design of an intelligent agent in a Human-Machine Interface (HMI) for the supervision of the filtering stage of a water purification plant, characterized by the ability to collaborate with the other agents that make up the entire plant. For the projection and design of the system, the Unified Modeling Language (UML) and Petri nets (PN) are used for the simulation and validation of the system, and the implementation of the agent from macros in C language, starting from a methodology of multi-agent design that is applied in this document. The implementation of the intelligent agent in an HMI associated with multi-agent architecture, which allowed to evaluate its behavior through the analysis of the properties of the PN and experimental tests, demonstrating the correct operation of the device, response times and its dynamic behavior based on of the functional requirements of the water purification plant and comparisons with similar works.

Keywords. Multi-agent systems, automation systems, Human-Machine Interface, intelligent agent.

1. Introducción

With the advancement of technology and new production requirements, the processes for improving the quality of products or services needs to be optimized [1]. Water purification plants are no exception, where the implementation of smart sensors and other autonomous devices are part of the continuous improvement of these plants, reducing failures and stops generated by damage or unplanned maintenance, which alter the operation of the system. [2, 3, 4]. For this reason, it is necessary to know each of the phases of the processes that are executed, their synchronization and collaborations. The modeling of the system is essential to define details of implementation, sequence of operations and their interactions, which guarantee compliance with the functional requirements, even more so when distributed automation architectures are used, which are more susceptible to structural and functional conflicts [4, 5].

SCADA systems in drinking water treatment plants are very useful as supervision and command elements, but in some cases, it is necessary to add other means of local supervision with less privileges such as Human Machine Interface (HMI) [6]. These must be adequately integrated with the different subsystems and devices that make up a distributed architecture to efficiently take advantage of the available technological resources and that adequately respond to the functional requirements of the system [4, 7, 8, 9]. The distributed architectures that have been implemented solve problems that cannot be solved using traditional automation [10, 11], but they still have challenges mainly related to their projection and integration, which can be overcome using appropriate methodologies and tools. [12]. Intelligent agents offer an adequate solution for these inconveniences, where control problems are distributed in different agents, which act autonomously, and can collaborate with each other to solve more complex problems of the plant [13, 14, 15] .

In the present work we propose the design of an intelligent agent housed in an HMI, as an element of a multi-agent system of a water purification plant, which, based on programming in C language, can perform communication and collaboration with other system agents using a reactive architecture. For the design of the agent and its architecture, a methodology based on Unified Modeling Language (UML) is used [15, 16] and for the validation and verification of the properties of the system, Petri nets (PN) are used [17, 18].

2. Automation System Analysis

In this section the analysis of the proposed system is carried out, starting from a general architecture of the water treatment plant, to contextualize the functions, roles and collaborations of the Intelligent Agent housed in a Human Machine Interface (A-HMI) within the system. After this phase, the A-HMI analysis is developed, from its description, functional requirements, and the architecture of the agent.

2.1. General System Architecture

The general automation architecture of the proposed water purification plant is shown in figure 1, within which the A-HMI is included. The architecture shows the level where the A-HMIs are located within the system, which are designed to be integrated into the intelligent distributed control system, where it is necessary to define a modular design with communication networks from the field level to the level of supervision, to ensure distributed intelligence [19, 20]. The communication of the different devices at the control level is based on the Profinet protocol and to satisfy the requirement for intelligent instrumentation at the field level, a MODBUS RTU communication network has been arranged [5]. Figure 1 shows the arrangement of the different devices of the system and their interconnections, made up of programmable automatons (PLC) that execute the intelligent algorithms in the distributed control system and at the lower level are the intelligent sensors. Regarding the supervisory system, two Kinco HMI panels have been arranged, each one of them houses an A-HMI, and they are used to visualize and monitor the dynamic behavior of the filters, the backwash tank, and the wash tank. For the development of advanced automation features, the intelligent supervisory system uses SCADA CODESYS, which is executed from a computer with wireless connectivity.

2.2. HMI-based System Analysis.

The analysis of the proposed system considers the different functional requirements, the control architecture of the system, as well as the different actors involved in the control system. These system actors can be a person, group of people or a machine that interacts with the system externally or internally [9]. Next, all the system actors shown in figure 1 are listed, which can be real or virtual: Operator (Op), Plant Manager (JP), Automation Specialist (EA), Decision-making algorithm in the AS-I sensors (ATD-S), Decision-making algorithm in AS-I communications processor (ATD-AS-i), Decision-making algorithm in the programmable logic controller (ATD-PLC) and Decision-making algorithm in the HMI (ATD-HMI). In this last actor, (ATD-HMI), its projection, design and evaluation are analyzed, to establish cooperation with the other actors of the system shown in figure 1, and that they are part of a multi-agent system.

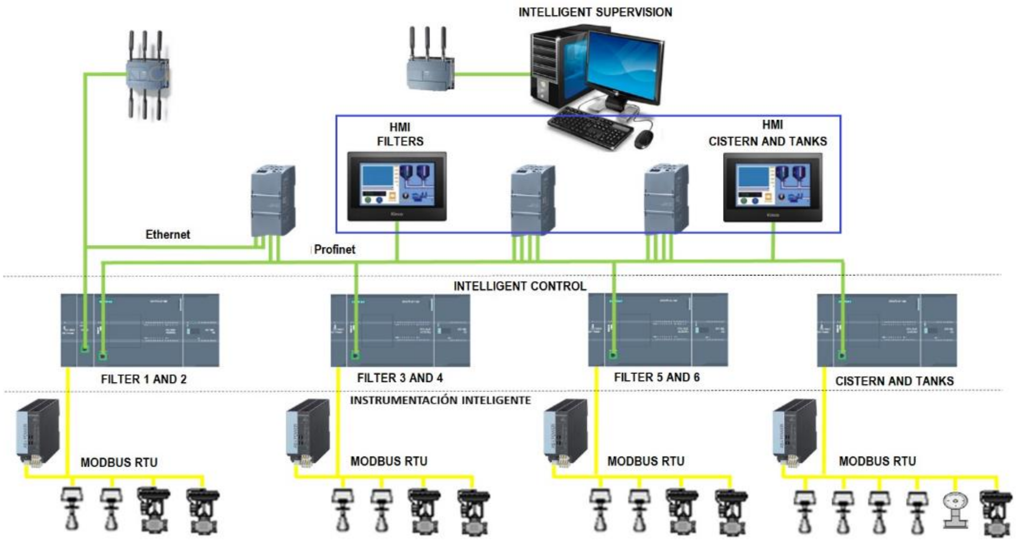

Figure 1. General Architecture of the Integrated Automation System

2.2.1 Functional Requirements

Functional requirements are invariable conditions that must be met by different systems or equipment. For example, the functional requirements in the management, supervision and control from the A-HMI are given by:

- Manage Users.
- Manage Process Monitoring.
- Manage working modes and desired values of Process parameters.
- Manage alarms.
- Manage historical data

In addition to the functional requirements, it is necessary to define the non-functional requirements, which are properties or qualities necessary to guarantee the correct operation of the system to meet its functional requirements [21]. The non-functional requirements of the A-HMI are:

- Screen size: Minimum 7 inches of color graphic screen.
- 128 MB of ROM memory and 512 KB of RAM memory.
- Communication via Profinet (Ethernet 10/100 Mbps)

Considering the functional and non-functional requirements, and the security requirements, the modular structure of the integrated automation system is proposed, which is shown in figure 2, where the arrangement of the two HMIs of the system and its components can be seen.

2.2.2 Security Requirements

The security requirements were considered in the modular structure of the Integrated Automation System that is illustrated in figure 1 and contribute to guaranteeing the security of the information of the system [22].

- Confidentially, allows only authorized users to access the system.
- Integrity, protects the data against the alteration or corruption of the information.
- Availability, that the design guarantees robustness and enables fault tolerance.

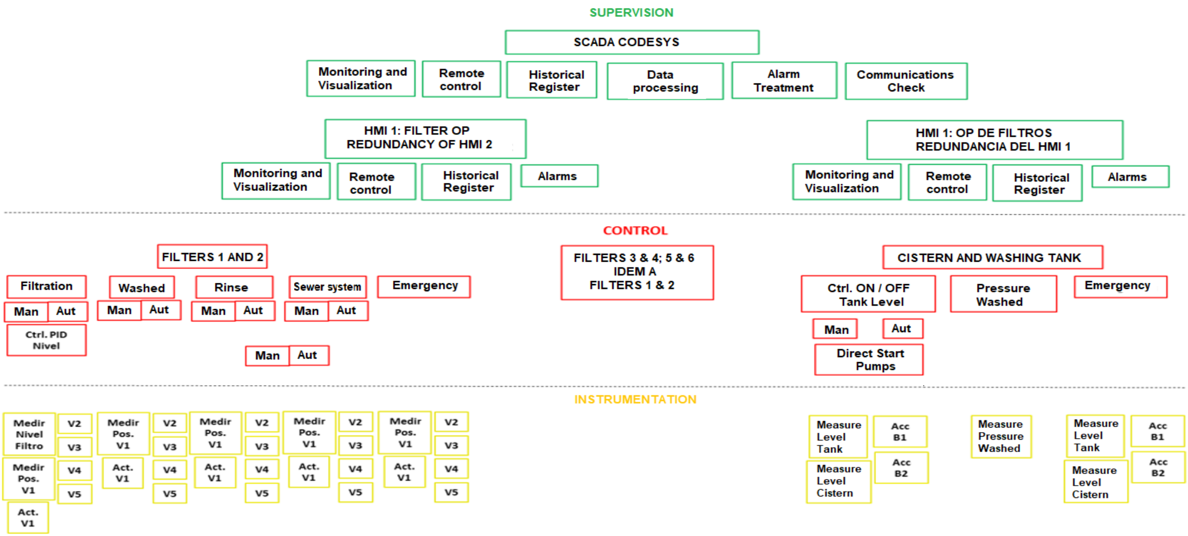

Figure 2. Modular structure of the Integrated Automation System

3. System modeling with UML and Petri Nets

For A-HMI modeling, a design based on a multi-agent system has been considered. This paradigm of agents is being widely used as an approach to modeling control systems and industrial developments, due to the decentralized nature of the problems in these areas [23], and the problems generated by business and manufacturing environments [24, 25]. Therefore, for the proposed case study, the use of UML has been considered for the design of the A-HMI. Through this graphic language it is possible to visualize, specify and document each of the parts of the A-HMI design [26, 27].

3.1. Use Case Diagram

Table 1 presents a description of the monitoring and control use case from the A-HMI and figure 3 shows the use case diagram.

Table 1. Description of the use case monitoring and control from the A-HMI

Use Case	Manage supervision and control from A-HMI
Description	Performs supervision and control from A-HMI
Pre-condition	HMI communicates with PLC and works as intended
Actors	ATD-HMI, Op, JP, EA
Failure Conditions	HMI fault or stop occurs

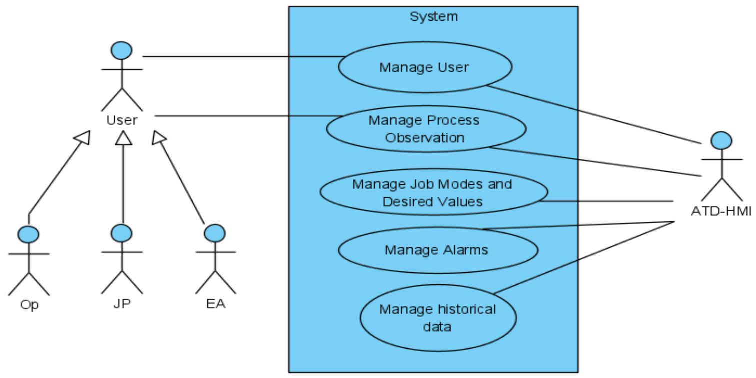

Figure 3. Supervision and control use case diagram from A-HMI

3.2. Activity Diagram

UML activity diagrams basically model the flow of activities and can be adapted to object-oriented programming [21]. The use of UML as a tool for system modeling is an important aid in the description of the functional requirements of the system [27, 28], although it does not allow describing the dynamics of the process. Next, in figure 4, the activity diagram corresponding to the use case described in section 3.2 is shown.

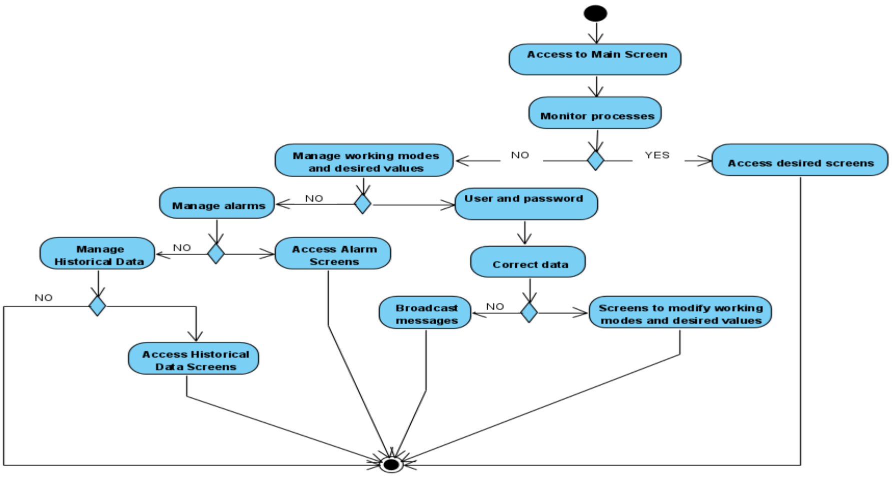

Figure 4. Activity diagram of supervision and control from the HMI

3.3. Sequence Diagram

Sequence diagrams show the interaction of the set of objects or actors involved in the process. This sequence diagram contains the implementation details, including the objects and classes that are used to implement the scenario and the channels of the messages exchanged between the objects [29, 30]. Figure 5 shows the sequence diagram of the A-HMI.

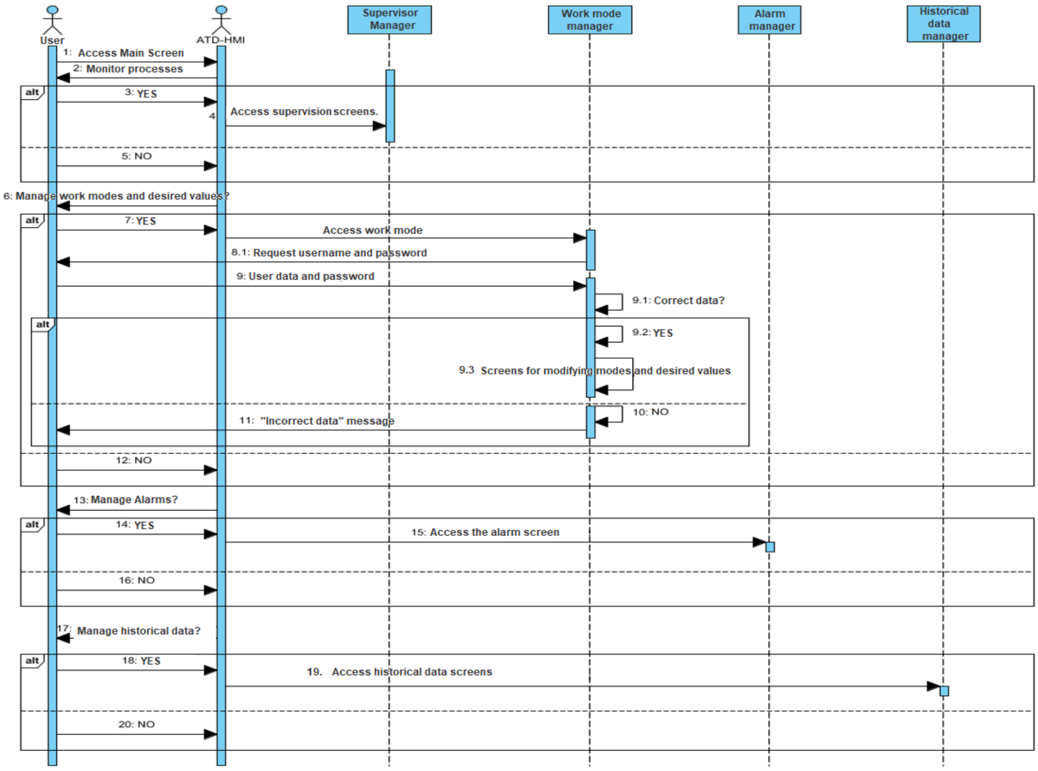

Figure 5. Sequence diagram of supervision and control from HMI

3.4. Modeling with Petri Nets

Once the A-HMI modeling for the control and supervision of the filtering process subsystems has been developed, the Petri net of the activity diagram is constructed. Petri nets make it possible to determine if the elaborated models are well conceived, for which it is necessary to ensure that they comply with the main functional properties of Petri nets; This will allow to obtain the verified and validated models through simulation. The models in Petri nets (PN) have many properties, of which the properties of Vivacity, Reachability, Limitation and Reversibility have been considered, which guarantee the functionality of the system [18, 31]. Figure 6 shows the Petri net of the activity diagram in figure 4, where the Visual Object Net ++ software has been used [20].

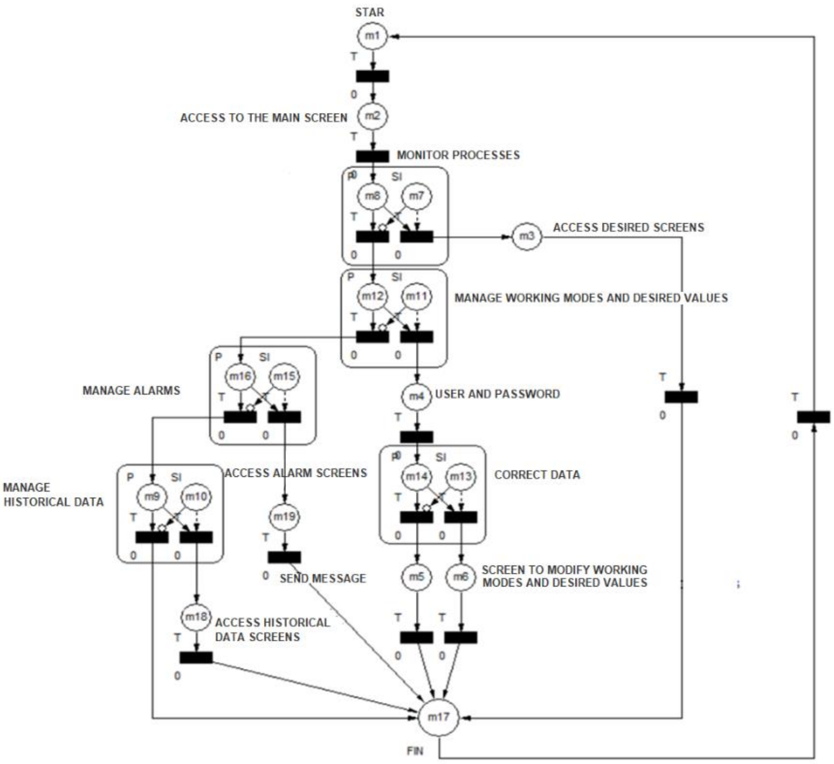

Figure 6. Petri net for supervision and control from A-HMI

4. Analysis of Results.

Once the modeling in UML and Petri nets has been carried out, the analysis of the obtained models and their influence on the obtained results is carried out. The models elaborated in UML contribute to the design of the A-HMI the characteristics from the approach of each diagram, however, the UML diagrams represent the behavior of the system in a static way, therefore, the translation of the model to Petri nets is necessary to determine the dynamic behavior of the system.

The sequence diagrams have allowed to obtain a general vision of the system, identifying all the functionalities present in the A-HMI [32, 33] and the interactions with the actors of the general process and of the A-HMI, considering a time base. The sequence diagrams are generated to determine the objects that are required in the implementation of the A-HMI over time, which clearly defines the roles of the objects in the process flow of the filters of the water purification plant. The activity diagrams represent the dynamic part of the UML, showing the flow of the processes carried out by the A-HMI. In addition, they allow activities to be carried out in parallel or other alternate flows that occur within the activity, thus achieving to define the behavior of the process to be carried out. The graph shown in figure 4 shows is the flow of information between the different screens of the A-HMI and from which the corresponding Petri net was obtained. For reasons of space, the diagrams of each screen and of the subprocesses have not been included, but the general activity diagram establishes the flow that is generated in the A-HMI.

Finally, with the Petri net of figure 6, the behavior of the system is analyzed through validation (compliance with the functional requirements of the system) and verification (compliance with the structural and behavioral properties of the network). The Petri net has been developed in Visual Object Net ++ and PIPE program [34, 35, 36]. The latter has mathematical analysis tools to evaluate the properties of the Petri net. In the Petri net analysis of the figure 7 the two properties mentioned at the beginning of this section have been considered. The Vivacity property is verified through simulations and allows determining that the networks are kept in continuous movement through the different loops and states of the processes. Additionally, the subroutines have been disactivated and activated so that the system automatically starts executing the selected variant. Through the PIPE software, the invariant equations T (transition) and P (place) are obtained. The solutions of the place invariants have all their elements between zero and one, which means that all the elements of the network belong to sets where the number of tokens is conserved, which is very important from the control point of view automatic since it guarantees, among other aspects, the property of limitation. The transition invariants T also their elements between zero and one, indicating that the number of shots of each transition is finite and different from zero, that it will not remain in an infinite cycle (free of locks) and that they all participate in the entry / exit paths (there are no sources or sinks), which implies the fulfillment of the liveliness of the network. In this way, it is demonstrated that the model complies with the Vivacity property, because the network does not have locks that limit the operation of the network.

Petri net invariant analysis results

T-Invariants

T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18
1	1	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
1	1	1	0	0	1	0	0	1	0	1	1	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	1	0	1	0	1	0	1	0	0	0	0
1	1	1	0	1	0	1	0	0	0	1	0	0	0	0	0	1	0	1
1	1	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	1	0

P-Invariants

P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P-Invariant equations
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	M(P3) = 0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	M(P5) = 0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	M(P7) = 0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	M(P11) = 0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	M(P14) = 0

Figure 7. Result of PIPE: Matrix of invariants T and P of the Petri net

The reachability property has also been analyzed through simulations in the network, since it is possible to determine that all the states of the network are reached at a certain moment. As in the Vivacity property, subroutines have been disactivated and activated in the process to determine if all the states are reached during the execution of the system. Figure 8 shows the matrices obtained with the PIPE software, where through the matrices for marking and enabling transitions, it is observed that the network is binary and that its states will always be finite or zero in the initial matrix. Using the state space analysis tool, the PIPE software demonstrates that the Petri net is unconstrained, secure, and free of locks, which corroborates the analysis performed in the Visual Object Net ++ software.

Marking

	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Current	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Enabled transitions

T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18
yes	no	no	no	no	no	no	no	no	no	no	no	no	no	no	no	no	no	no

Petri net state space analysis results

Bounded	true
Safe	true
Deadlock	false

Figure 8. PIPE Results: Petri Net States, Transitions, and Marking Matrices

The implementation of the supervisory system in the A-HMI is carried out using the Kinco HMIware program in its version 2.5, which has an interface to be able to carry out the programming, and depending on the operation of the Petri nets, the necessary windows of the proposed models. In figure 9 you can see the HMI Kinco MT443TE screen, with its development interface the HMIware software, connected through a USB connection, and powered at a voltage of 24 V. Figure 10 corresponds to the user session, the which can access the five options available to the system. Similarly, in the lower right part of the screen, there is the START button, which allows you to return to the initial screen in the system, closing all types of access to prevent other unauthorized actors from entering.

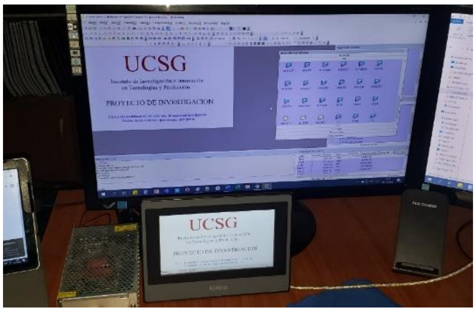

Figure 9: Software Kinco HMIware

Figure 10: Tab selection window

5. Conclusions

The general modeling of the system using UML for the A-HMI, allows to corroborate the usefulness of the UML modeling to specify, design and document the water filtration system in a precise way, describing its context and defining the functional requirements. In addition, it has made it possible to specify the actors or components, describe the services or use cases and define the relationships between the actors. The use of modeling based on UML-PN for the A-HMI has been used successfully in the projection, design, and modeling of the system, allowing to verify the properties and validate the functional requirements defined for each one of the implemented components, which guarantees the correct simulation of the models, generating stable, functional and robust systems.

The architecture proposed for the intelligent supervisory system is a hierarchical modular distributed architecture, which allows the control of the different levels of automation, as well as the command and monitoring of the water filtration process. The implementation of the A-HMI in a Kinco HMI demonstrates the physical operation of the supervisory system, demonstrating its flexibility and modularity, which has been projected from the UML modeling and has been validated and verified through the analysis of the properties of the control networks Petri, concluding that, based on the methodology used, an optimal functioning system can be achieved from the UML-PN modeling.

References

[1] Vilanova, R., Santín, I., & Pedret, C. 2017. Control en estaciones depuradoras de aguas residuales: Estado actual y perspectivas *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 14, no 4, pp. 329-345.

[2] Calderón Córdova, C. et al., 2018. Monitoring and automation of the water pumping and storage process applied to a water treatment plant. *13th Iberian Conference on Information Systems and Technologies (CISTI)*, 2018, pp. 1-6, doi: 10.23919/CISTI.2018.8399292.

[3] Campos, C. A. V., & Bravo, E. C., 2013. Técnicas de inteligencia computacional aplicadas a modelos de estimación de coagulante en el proceso de potabilización de agua. *Revista Facultad de Ingeniería Universidad de Antioquia*, vol. 69, pp. 205-215.

[4] Nuñez, J. R., et al., 2019. Tools for the Implementation of a SCADA System in a Desalination Process. *IEEE Latin America Transactions*, vol. 17, no. 11, pp. 1858-1864, DOI: 10.1109/TLA.2019.8986424

[5] Mendoza, E., Andramuño, J., Núñez, J., & Benítez, I. 2021. Deliberative architecture for smart sensors in the filtering operation of a water purification plant. *J. Phys.: Conf. Ser.*, vol. 1730, no. 1, p. 012088. doi: 10.1088/1742-6596/1730/1/012088.

[6] Valdez, J., Pandolfi, D., & Villagra, A. 2018. Red de sensores inteligentes para adquisición de datos de una planta de desalinización de agua. *Informes Científicos Técnicos-UNPA*, vol. 10, no. 2, pp. 83-95.

[7] Quezada-Quezada, J. C. et al., 2014. Diseño e implementación de un sistema de control y monitoreo basado en HMI-PLC para un pozo de agua potable. *Ingeniería, investigación y tecnología*, vol. 15, no. 1, pp. 41-50.
[8] Mendoza, E. et al., 2020. Network of multi-hop wireless sensors for low cost and extended area home automation systems. *Revista Iberoamericana de Automática e Informática Industrial*, vol 17, no. 4, pp. 412-423. https://doi.org/10.4995/riai.2020.12301
[9] Nuñez-Alvarez, J. R., Benítez-Pina, I., & Llosas-Albuerne, Y. 2020. Communications in Flexible Supervisor for Laboratory Research in Renewable Energy. *IOP Conf. Series: Materials Science and Engineering*, vol. 844, pp. 012016 doi:10.1088/1757-899X/844/1/012016
[10] Aguilar, J., Bolivar, A. R., Hidrobo, F., & Cerrada, M. 2012. *Sistemas MultiAgentes y sus Aplicaciones en Automatizacion Industrial*. 2nd Edition. Universidad de los Andes, Mérida, Venezuela.
[11] Zheng, Y., Wei, Y., & Li, S. 2018. Coupling Degree Clustering-Based Distributed Model Predictive Control Network Design. *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 4, pp. 1749-1758, doi: 10.1109/TASE.2017.2780444
[12] Wang, Z., & Wang, J. 2020. A Practical Distributed Finite-Time Control Scheme for Power System Transient Stability. *IEEE Transactions on Power Systems*, vol. 35, no. 5, pp. 3320-3331, doi: 10.1109/TPWRS.2019.2904729
[13] Sun, Q., Yu, W., Kochurov, N., Hao, Q., & Hu, F. 2013. A multi-agent-based intelligent sensor and actuator network design for smart house and home automation. *Journal of Sensor and Actuator Networks*, vol. 2, no. 3, pp. 557-588.
[14] Leitão, P., Rodrigues, N., Turrin, C., & Pagani, A. 2015. Multiagent system integrating process and quality control in a factory producing laundry washing machines. *IEEE Transactions on Industrial Informatics*, vol. 11, no. 4, pp. 879-886.
[15] Andramuño, J. et al., 2021. Intelligent distributed module for local control of lighting and electrical outlets in a home. *J. Phys.: Conf. Ser.*, vol.1730, pp. 012001. doi:10.1088/1742-6596/1730/1/012001
[16] Mekuria, et al. 2021. Smart home reasoning systems: a systematic literature review. *J Ambient Intell Human Comput*, vol. 12, pp. 4485–4502. https://doi.org/10.1007/s12652-019-01572-z
[17] Joyanes, L., & Zahonero, I. 2014. *Programación en C, C++, JAVA y UML*. Editorial Mc Graw Hill. 2nd Edition. México.
[18] Meziani, L., Bouabana-Tebibel, T., & Bouzar-Benlabiod, L. 2018. From petri nets to UML model: A new transformation approach. *IEEE International Conference on Information Reuse and Integration (IRI)*, pp. 503-510. doi: 10.1109/IRI.2018.00080
[19] David, R., & Alla, H. 2010. *Discrete, continuous, and hybrid Petri nets*, Editorial Springer, 2nd Edition.
[20] Micolini, O., Ventre, L. O. & Ludemann, M. Methodology for design and development of Embedded and Reactive Systems Based on Petri Nets. *IEEE Biennial Congress of Argentina (ARGENCON)*, 2018, pp. 1-7. doi: 10.1109/ARGENCON.2018.8646137
[21] Benitez-Pina, I. F., Lamar-Carbonell, S., da Silva, R. M., Eigi-Miyagi, P., & Silva, J. R. 2017. Design of automatic control system based on unified timed hybrid Petri net, *Dyna*, vol. 84, no. 200, pp. 80-89.
[22] Pressman, R. 2010. Ingeniería del Software Un Enfoque Práctico. Editorial Mc Graw Hill. 7th Edition. University of Connecticut, USA.
[23] Nuñez Alvarez, J. R., Zamora, Y. P., Pina, I. B., & Angarita, E. N. 2021. Demilitarized network to secure the data stored in industrial networks. *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 1, pp. 611.
[24] Hernández, M. P., Mcfarlane, D., Parlikad, A. K., Herrera M., & Jain, A. K. 2021. Relaxing Platform Dependencies in Agent-Based Control Systems. *IEEE Access*, vol. 9, pp. 30511-30527, doi: 10.1109/ACCESS.2021.3059273.

[25] Chao, Z., Weidong, Z., Yong, W., Danli, W., Hangbing, Z., & Xiaoya, D. 2019. Can Manufacturing Service Value Co-Creation Improve Business Performance in the Internet Era? *IEEE Access*, vol. 7, pp. 120360-120374, 2019, doi: 10.1109/ACCESS.2019.2937879.

[26] Kang, H., Li, J., Zhang, L., Feng, T., Yin Q., & Liu, Y. 2020. Agent-Based Decentralized Grid Model. *IEEE Access*, vol. 8, pp. 102595-102606, doi: 10.1109/ACCESS.2020.2999490.

[27] Bazyd\lo, G., Wojnakowski, M., & Wiśniewski, R. 2018. The use of UML and Petri net for graphic specification of the reconfigurable logic controllers. *AIP Conference Proceedings*, vol. 2040, no. 1, pp. 080004. https://doi.org/10.1063/1.5079138

[28] Yang, Y., Ke, W., Yang J., & Li, X. 2019. Integrating UML With Service Refinement for Requirements Modeling and Analysis. *IEEE Access*, vol. 7, pp. 11599-11612, doi: 10.1109/ACCESS.2019.2892082

[29] Vani, M., Kumari, M. C., Priya, M. H., & Harika, N. 2015. An effective language for object oriented design-uml (unified modeling language). *International Research Journal of Engineering and Technology (IRJET*), vol. 2, no. 5, pp. 1212-1218.

[30] Booch, G. 2005. The unified modeling language user guide. Editorial Pearson Education India, 2nd Edition.

[31] Aparna, V. K. 2018. Exploration of Compact Sequence Diagram Using Hyperlink. *Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, 2018, pp. 1-4, doi: 10.1109/ICCUBEA.2018.8697567

[32] Silva, M. 2002. Las Redes de Petri: en la Automática y la Informática. Editorial AC Madrid. 1st Edition. España.

[33] Depke, R., Heckel, R., & Küster, J. M. 2001. Roles in agent-oriented modeling. *International Journal of Software engineering and Knowledge engineering*, vol. 11, no. 3, pp. 281-302, 2001.

[34] Nuñez, J. et al., 2020. Design of a Fuzzy Controller for a Hybrid Generation System. *IOP Conference Series Materials Science and Engineering*, vol. 844, pp. 012017. doi: 10.1088/1757-899X/844/1/012017

[35] Liu, S., Zeng, R., He, X. 2011. PIPE + - A modeling tool for high level Petri nets. *Conference: Proceedings of the 23rd International Conference on Software Engineering & Knowledge Engineering (SEKE'2011)*, Eden Roc Renaissance, USA, July 7-9, 2011

[36] Dingle, N. J., Knottenbelt, W. J., & Suto, T. 2009. PIPE2: A Tool for the Performance Evaluation of Generalised Stochastic Petri Nets (PDF format). *ACM SIGMETRICS Performance Evaluation Review*, vol. 36, no. 4, pp. 34-39.

[37] Bonet, P., Llado, C. M., Puijaner, R., & Knottenbelt, W. J. 2007. PIPE v2.5: A Petri Net Tool for Performance Modelling. 2*3rd Latin American Conference on Informatics (CLEI 2007)*, San Jose, Costa Rica, October 2007.